

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 May 2001 (10.05.2001)

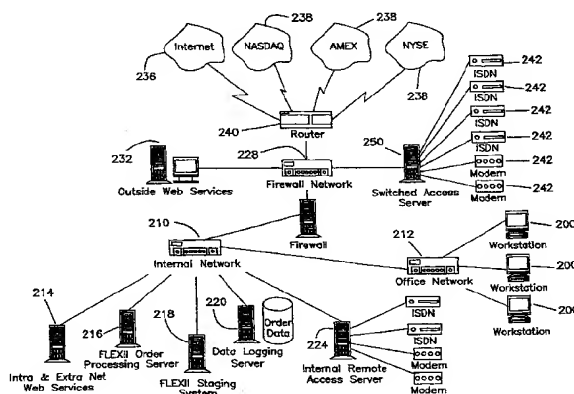
PCT

(10) International Publication Number
WO 01/33316 A2

- (51) International Patent Classification⁷: **G06F** (74) Agent: **BECKERS, J., Randall**; Staas & Halsey LLP, Suite 500, 700 Eleventh Street, Washington, DC 20001 (US).
- (21) International Application Number: **PCT/US00/41840**
- (22) International Filing Date:
3 November 2000 (03.11.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/433,659 3 November 1999 (03.11.1999) US
- (71) Applicant: **UNX, INC.** [US/US]; 175 E. Olive Avenue, Suite 200, Burbank, CA 91502 (US).
- (72) Inventors: **ABERNETHY, William, Randolph**; 175 E. Olive Avenue, Suite 200, Burbank, CA 91502 (US).
OLMEDO, Oscar; 175 E. Olive Avenue, Suite 200, Burbank, CA 91502 (US).
- (81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
— Without international search report and to be republished upon receipt of that report.

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR TRADING USER-DEFINABLE BASKETS OF FUNGIBLE GOODS SUCH AS SECURITIES



(57) Abstract: A system and method is disclosed for creating and trading *ad hoc* baskets of fungible goods such as securities. The invention permits users to purchase groups of species of fungible goods by simply inputting the target amount to invest and a list of such species such as stock symbols. The invention then effectively instantaneously (from the user's perspective) seeks out the current trading price for each species (e.g. current stock trading price) and computes the number of units (e.g. shares) of each species to be purchased in order to invest a total amount approaching as close as possible (but no greater than) the total target investment in the particular basket of species of fungible goods (e.g. stocks) specified. A means for dynamically "balancing" and "rebalancing" a basket is also disclosed which permits the user to change the weight applied to each species when determining the number of units (shares) to be traded. Each "basket" may be named by the user, is identified with a unique code assigned by the system, and is referred to as an "Individual Fund™" or an "IF". An Individual Fund may be traded as a unit much like a "mutual fund". The Net Asset Value (NAV) and total value of each Individual Fund is reported on a real-time or relatively frequent basis.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

**METHOD AND SYSTEM FOR TRADING USER-DEFINABLE BASKETS OF
FUNGIBLE GOODS SUCH AS SECURITIES**

5

Reference to Prior-Filed Applications

This application claims the benefit under 35 U.S.C. section 119(e) of our U.S. Provisional Patent Application Serial Number 60/110524 filed December 1, 1998 which is hereby
10 incorporated by reference in its entirety into this application.

Field of the Invention

The method and apparatus of the present invention relate to electronic trading of groups
15 of fungible goods such as securities. The present invention provides a computer-assisted system for creating, modifying, executing and tracking "baskets" of assets in real time. It allows individual investors to freely create, track and trade their own "funds" constructed and weighted according to their own criteria.

20 **Background of the Invention**

Electronic trading of securities and other fungible goods by both professionals and consumers has become the dominant means of trading such fungible goods. Moreover, the availability of direct consumer trading via Internet and other on-line services has grown dramatically in recent years. Additionally, the use of "mutual funds" as investment vehicles has
25 become commonplace for individual investors as well as institutional investors. However, the structure and management of each such "mutual fund" is under the exclusive control of the fund manager. Moreover, the net asset value of a customer's shares of a mutual fund is typically updated only once per day. It is recognized that consumers may desire to invest in specific groups of securities defined by parameters of their choosing. However, for an individual investor
30 (whether she be a professional or consumer) to purchase individual stocks meeting specific user-

definable criteria is extremely time-consuming and does not permit rapid adjustments in response to market conditions or changing strategies. It is seen as an advantage to enable trading entirely flexible "baskets" of securities rapidly with immediate reporting on the net asset value ("NAV") of the newly created asset.

5 Many tools are available to investors for "filtering" or choosing investments according to various criteria. For example, an investor may wish to purchase the ten airline stocks with the lowest price-to-earnings ratio, or the fifty best-performing stocks listed in the Standards & Poors Index. While the investor may do so on a stock-by-stock basis, a system for instantaneously implementing such a strategy with immediate access to execution and to net asset values is
10 desirable. In essence, the present invention permits individuals to define, trade, and manage their own "individual funds" consisting of "baskets" of securities chosen according to the criteria set by the individual investor. It further freely allows the user to weight individual assets and maintain plural "baskets." Additionally, where a security such as a stock is sold or "sold short" from one basket, shares for such a transaction may be moved from another basket owned by the
15 same user (if such shares are in such a basket) thus avoiding transaction fees. This also would reduce potential market impact, i.e. it has a stabilizing influence on the market rather than a destabilizing influence.

Description of the Related Art

Systems for reliably and safely transacting securities trades over electronic networks are
20 well known in the art. Currently, there are numerous on-line services providing various levels of stock trading over the Internet and proprietary networks. Brokerage houses have maintained their own proprietary networks for trading clients' investments. Some of these networks are so-called "ECN's" which permit cross-trading within the system. However, heretofore there has been no system implemented which permits rapid *ad hoc* creation, execution, and management
25 of individual "baskets" of investment vehicles such as stocks and mutual funds according to user-defined, modifiable criteria. All existing systems are inherently and internally organized to promote single issue trading, not basket formation and management. Unlike prior art, the present invention, which has been implemented under the mark "FLEX II™", is specifically and holistically organized, from end to end, as an "Individual Fund" (multiple issue) trading vehicle,
30 i.e. for trading baskets of fungible goods such as securities. Not only does the invention permit

users to establish a particular individualized basket (Individual Fund™ or “IF™”), but to balance and later re-balance its contents. In one implementation, for the purposes of balancing and rebalancing the contents of the basket, a graphic interface may be provided permitting the user to “drag” a pictorial bar on a displayed bar chart, each bar representing a particular issue in a “basket.” The pictorial bars for the remaining issues change in response to keep the “basket” balanced (i.e. totaling the target investment amount).

Brief Summary of the Invention

The present invention integrates computer hardware, software and communications networks to permit individuals or institutions to dynamically and instantaneously manage “baskets” of assets (fungible goods, especially securities). These baskets, termed “Individual Funds” may contain such assets as stocks, bonds, mutual funds, futures, options and the like. The investor may trade these “baskets” or “Individual Funds” as if they were single assets, may readily modify their contents (i.e., add or delete individual assets) and may instantaneously determine the Net Asset Value (“NAV”) of each basket. As specifically implemented, NAV is defined as the total value of the basket divided by the sum of the number of shares. Alternatively, some on-line services may deduct fees and costs. The core of the method and apparatus is the ability to create, modify, execute, confirm and track an asset portfolio that can be traded as a single unit. A user may therefore maintain a brokerage account containing one or more “baskets”, with the ability to trade each “basket” as a unity.

The present invention is a method and apparatus providing a highly reliable and stable financial trading system with which investors may define criteria of their choosing to select investments in groups forming Individual Funds. The best mode of the present invention combines technologies in such a way as to preserve data integrity and produce high levels of availability and security, while insuring both privacy and continuous availability. It also provides for distribution of processing burden and incremental system expansion within a scaleable framework. Components are clustered to permit dynamic load balance. The best mode of the present system further permits essentially unlimited scalability and seamless integration with any network, proprietary or otherwise. By using a messaging architecture for the software to demark the boundary between the internal behaviors and external system stimuli, the present

invention allows unrestricted flexibility with regard to implementation vis-a-vis exchanges quotes and users.

The present invention may also be viewed as an order management and executing network designed to execute instantaneous multi-issue "basket" transactions. The essence of the present invention is a set of distributed software components. Each component maintains all persistent data ("state information") within a redundant transactional framework. This component-based transaction system allows system elements to be distributed throughout a network of relatively low-cost clustered multiprocessor computers. All transactions are completed atomically (as an individual unit) by collecting Exchange Interactions, Broker/Dealer Interactions, Electronic Communications Network ("ECN") interactions, database logging and system business logic into transactional component objects. ECN's (which are well-known to those skilled in the art) are typically implemented as independent order books for one or several markets. By decomposing the core services into business logic components that can run on several system processors concurrently, efficiency, performance, scalability and reliability are optimized.

Normally, in its best mode of implementation, the invention is facilitated by a redundant primary network which is backed-up by an off-site catastrophic failure contingency network. The baseline infrastructure is easily extended to support numerous multiples of predicted traffic levels. All network links are star wired and redundant allowing for easy, zero-impact repair in case of channel failure. All hubs and routers are equipped with fully redundant power supplies and back-planes. Internal networks are protected by a firewall. The firewall system is clustered for fault tolerance and implements proxy and reverse proxy services. Primary outside links currently operate over redundant frame relay networks and may be backed up by multiple dial-up links, such as ISDN. Primary and back-up networks should always be supplied by different carriers and technologies each supplying more than ample bandwidth for full volume trading. The present invention as implemented operates in a tightly integrated Component Object framework built on top of Windows NT™. The invention could also be easily implemented using CORBA on UNIX-based systems, or other component frameworks implemented on other operating systems. Mission critical clustering and transaction management technologies are also implemented. Data services are managed through industry standard Structured Query Language

("SQL") databases. The combination of standards-based technology provides an advanced and stable development environment. Microsoft's Windows NT™ offers a coherent set of services for modern scalable component system deployment. However, other operating environments such as UNIX or LINUX could readily be used. SQL databases allow *ad hoc* queries to generate unique and timely data perspectives. The best mode implementation of the present invention uses components that are designed to take advantage of 64-bit technology and massive primary and secondary storage facilities. Deployment of the best mode implementation may take place on fault-tolerant clusters of standard multi-processor PC systems. Redundant Array of Independent Disks, Level Five ("RAID 5") disk systems are shared between clustered machines.

The present invention is a fully distributed system allowing processing power and network infrastructure to be added incrementally as demand increases. A person skilled in the art will readily recognize that a wide set of variations in choice of hardware and software is possible. New reliable data storage devices will undoubtedly evolve over the coming years and may be readily interchanged for the specifically-implemented modes described herein. For example, mass storage in the form of solid state devices could easily be substituted for hard drives as the cost and reliability of that technology changes. Further, the precise operating systems, computer languages and database technologies described herein are not critical to the implementation of invention. The claimed invention may be built upon a completely different platform and yet operate exactly as claimed.

Brief Description of the Drawings

Figures 1A through 1E are interrelated flow charts ("event traces") illustrating the invention as specifically implemented by the inventors and believed to be the best mode of implementing the claimed invention.

Figure 2 is an example conceptual diagram of the hardware interrelationships of the invention as implemented in a particular architecture.

Figure 3 is a diagram of the core database schema implemented in the best mode of the present invention.

Figure 4 is an example of a user interface implementing dynamic basket balancing using graphical and/or numeric input.

Figure 5 is a sample user report summarizing a basket trade.

Detailed Description of the Invention and the Best Mode

The present invention represents an integrated computer hardware/software/network system implemented to permit reliable dynamic trading of baskets of fungible goods.

5 Fundamentally, the core of the present invention is a component-based order management object set tailored to the task of multiple issue "basket" trading. While the method and system claimed can be implemented in such as way as to "overlay" existing trading systems, the best implementation is a ground-up architectural structure built around, and based upon the concept of basket trading. Additionally, core components should be designed to optimize scalability,
10 readily consuming multi-processor resources and operating efficiently in collaboration scenarios across high speed data networks.

In its preferred embodiment, the present invention operates under Microsoft Transaction Server ("MTS") transaction control and intelligently fails over to secondary systems in the event of primary system failure. There are alternatives to MTS which are well-known to those skilled
15 in the art and may be freely substituted. Object services are extended to consumer applications for *ad hoc* spreadsheet analysis, charting and text-based reporting. Web extension DLLs provide user interface integration with core object oriented components and allow custom interfaces to be rapidly produced for the changing needs of institutional and professional users. All data services are accessed through component interfaces allowing every aspect of system order processing to
20 be tracked in a persistent data store.

Order processing components enable Internet and integral client applications to enter and manage market and stop orders. All orders are transactional and logged in the data store. Orders can be created, queried and canceled through a set of components that run through a load balancing front-end on several back-end systems. Orders are queued to the order router and
25 logged in the database within a single transaction.

The best mode of the present invention includes a logging system in which all orders are tagged with customer identification numbers and logged in a fault tolerant relational database. Orders are time-stamped at inception and at time of execution. This logging system supports dynamic data retrieval through various client interfaces. The present invention's data store
30 facilitates historical reporting, research-oriented tasks, regulatory compliance and surveillance.

Additionally, the best mode of the present invention incorporates a dynamic order execution service ("OES") interface framework. OES interface objects can be activated and deactivated on the fly to allow on-line OES field replacement and multiple concurrent execution providers. Orders are processed using messaging and transactional queues. The order routing service ("ORS") forwards orders to the most effective OES based on load and fill rate. The OES may execute the order through its attached exchange, writing an execution message to the confirmation service. If the OES cannot execute the order the order is queued back to the ORS to receive further routing. All messages are moved within a transaction ensuring that system and/or network failures cannot cause data loss.

Figures 1A through 1E are event traces (flow charts) illustrating the overall function of the present invention as implemented by the inventors. More specifically, Figure 1A represents the sequence of events which take place to effectuate a user login. The user submits a unique user identification ("ID") and password 10 to the network interface object ("CF2WWWExt") 14 (in this instance, the interface is with the worldwide web) which requests authorization from the authentication object ("CAuthentication") 16. The submitted ID and password are compared with the proper ID and password 18 stored as part of the core database ("F2CoreDB") 20. The results of the comparison are returned 22 to the authentication object 16 which notifies 24 the network interface object ("CF2WWWExt") 14 of the results of the comparison 18. If the user's ID and password are authenticated (i.e. match those stored in the core database) then a new user session and session key is created 26 by session object ("CFlexIISSession") 28 and a new session record is written 30 in the session database ("F2SessionDB") 32. The session record proves the user is currently logged-in, evidenced by a database session key matching that embedded in the user's form. Failure to authenticate blocks the user from continuing. Contained in the session record is also a contact (user) ID and current account ID. Upon completing writing a new session record 30 the Session Object 28 and the network interface object 14 are notified. The network interface object ("CF2WWWExt") 14 then issues a command to "get authorized accounts" 36 to the authentication object ("CAuthentication") 16 which checks the core database 20 ("F2CoreDB") to determine which accounts are authorized for the particular user to access 38. The accounts which the user is authorized to access are returned 40 to the authentication object 16 which, in turn, communicates them to the network interface object ("CF2WWWExt") 14.

The accounts which the user is authorized to use are then displayed to the user on an "account selection page" 42. The embedded (hidden) session key is returned with the Page 42.

Figure 1B illustrates the sequence of events which next take place for the user to select an account. Specifically, the user (who has received authorization) selects an account from the displayed form 42. The selection is transmitted to the network interface object ("CF2WWExt") 14 which executes a command to lookup the unique session identifier embedded in the user's account-selection form 54. The session record 30 earlier created is then retrieved 56 from the session database ("F2SessionDB") 32 and returned to the session object ("CFlexIISession") 28 which then communicates the session ID to the network interface object ("CF2WWExt") 14. The network interface object 14 then requests an account summary 58 from the account information object ("CAccountInfo") 16 which looks up the account record and all "open positions" 60 in the core database ("F2CoreDB") 20. That raw data are returned to the account information object ("CAccountInfo") 16 which calculates the account statistics 62 (e.g. buying power, account balance, special memorandum account, etc.) and returns the results of that calculation to the network interface object ("CF2WWExt") 14. The network interface object ("CF2WWExt") 14 then requests 64 summaries of each Individual Fund™ ("IF") from the account information object ("CAccountInfo") 16 which looks them up 66 in the core database ("F2CoreDB") 20. The raw data of each record associated with the unique user identification number are then sent from the core database ("F2CoreDB") 20 to the account information object ("CAccountInfo") 16 which computes a summary of the contents of each IF, including current and purchase NAV, and current and purchase total IF price. The results of that computation, a summary of each IF 68, is communicated to the network interface object ("CF2WWExt") 14 and displayed 70 to the user with the hidden (embedded) session key sent with the displayed account summary.

Figure 1C illustrates the step in which the user enters a list of symbols representing species of fungible goods (e.g. stocks) to purchase and a total target investment amount. Specifically, the user enters a list of symbols and total dollar amount for a particular IF 76 and sends it to the network interface object ("CF2WWExt") 14 which looks up 77 the client session using the unique session key embedded in the form used by the user to look up the session, the network interface object ("CF2WWExt") 14 queries the session object

("CFlexIISession") 28 which, in turn, queries the session database ("F2SessionDB") 32. The results of the query are returned to the session object ("CFlexIISession") 28 which, in turn, returns the results to the network interface object ("CF2WWWExt") 14. In order to determine how many units (shares) of each species (stock symbols) need to be purchased, the real-time ask price for each species must now be determined. (Note: Figure 1A through 1E illustrate a "buy" cycle. A "sell" cycle would be obvious to one skilled in the art based upon the illustrated "buy" cycle illustrated.) Thus, the network interface object ("CF2WWWExt") 14 submits 77.5 the list of symbols the user entered to the preview object ("CPreview") 78 which, in turn, requests quotes 80 from a local quote consumer object ("CQuoteConsumer") 84. The local quote consumer object ("CQuoteConsumer") 84 then calls 86 a remote quote server ("CQuoteServer") 88 which retrieves real-time quotes 90 from a quote feed database ("PQPPcQuote") 92. The real-time quotes are sequentially communicated back to the quote server object ("CQuoteServer") 88, the local quote consumer object ("CQuoteConsumer") 84 and to the preview object 78. Then, a preview of the proposed IF composition is returned 96 to the network interface object ("CF2WWWExt") 14 and displayed 98 to the user for her approval. Note that quotes obtained from the remote quote server 88 may be transmitted via a wide variety of mechanisms including telephone lines, satellite links, radio transmissions, dedicated hard-wired connections, Internet links and/or equivalent or similar mechanisms.

Figure 1D models the next step in use of the present invention in which the user approves the previewed IF and executes multiple trades nearly simultaneously to create an IF. Specifically, the user submits 110 the preview form she has reviewed and approved to the network interface object ("CF2WWWExt") 14 which look up the session 112 in the session object ("FlexIISession") 28 which, in turn, fetches the session record 114 from the session database ("F2SessionDB") 32 and returns it in sequence to the session object ("CFlexIISession") 28 and the network interface object ("CF2WWWExt") 14. The contents of the approved IF are then sent by the network interface object 14 to the order object ("COrder") 120 which creates an IF order transaction. The order object ("COrder") 120 then checks for buying power from the account information object ("CAccountInfo") 121 and returns a confirmation or rejection to the order object ("COrder") 120. If any step in the transaction is not completed, the entire transaction is aborted. The order object ("COrder") 120 then writes the order 122 to the core

database ("F2CoreDB") 20 which, in turn, returns a confirmation 124. Orders for each component of the IF are then queued to the execution queue ("FlexII_Ex") 126 which returns confirmation of entry into the queue to the order object ("COrder") 120. When all orders are successfully queued, the order transaction commits. The order object 120 returns the unique IF order ID and returns 128 the confirmation and IF order ID to the network interface object 14. The order ID is displayed to the user 130.

Figure 1E models the steps which take place to execute the orders to create a basket entered and approved by the user as described above. Specifically, the execution router ("CExecutionRouter") 140 retrieves asset orders from the execution queue ("FlexII_Ex") 126. The execution router queues the order 146 to an order execution service ("OES"), choosing each service based upon current performance criteria such as speed of execution. The execution router then waits for the next order to arrive in the execution queue 126. Thus, the orders placed by users and the execution of those orders are decoupled through means of asynchronous messaging. The execution router iterates through the foregoing steps perpetually routing all orders as they arrive.

Each order execution service has its own input queue ("FlexIIExOES") 142 and its own pending queue ("FlexIIExOESPending") 144. The order execution receiver ("OESQReceiver") 146 validates the incoming order message and forwards it to the order execution service's internal pending queue. The order execution service network manager ("POESMISLD") 148 (which is specific to each order execution service) retrieves the pending order message and submits it to the external service 150 (e.g. Island ECN). Note, the illustrated order execution service network manager is a "plug-in" for the Island ECN. The external service 150 responds with an execution report 152 to the order execution service network manager 148 which removes the pending order from the OES pending queue 144 and forwards the executed order to the confirmation service queue 154. The order execution services iterates through the foregoing steps perpetually executing all orders and forwarding confirmations.

The confirmation object ("CConfirmation") 160 retrieves the confirmed order from the confirmation queue ("FlexIIConf") 154, updates the core database ("F2CoreDB") 20, and then e-mails 162 an execution report to the contact (user) originally responsible for placing the order.

The confirmation object 160 iterates through the foregoing steps perpetually confirming all in-bound orders to the core database 126 and the associated contact.

Figure 2 illustrates the physical network architecture of the present invention as implemented. Specifically, an internal network 210 connects an office network 212 and several
5 servers (Intra and Extra Net Web Services Server 214, Order Processing Server 216, Staging System 218, Data Logging Server 220 and Internal Remote Access Server 224) through a firewall network 228 to outside computers and networks. These outside computers and networks include outside web services 232, the Internet 236, commodities exchanges 238 (which are shown connected to the firewall network 228 via a router 240), and users' computers which
10 connect via various telecommunications modems 242 through a switched access server 250.

Figure 3 is the core database schema. According to standard, well-understood database theory, Figure 3 illustrates a normalized relational database. Each rectangle represents a normalized table containing several fields, one or more of which are designated as the table's primary key. The relationships in this database are modeled with connecting lines such that each
15 line depicts a key pointing to the table containing the primary key in the relationship.

For example, each Account 252 contains a CustomerID 253 which is a foreign key corresponding to a primary key "ID" 254 in the Customer table 255. Thus, multiple account records may be associated with a single customer record. A primary key may also be a concatenated key such as in the AccountContact table 256. Note that according to the database
20 structure illustrated, multiple contacts 257 may control a single account 252, and a single contact 257 may control many accounts 252. Similarly, one IF 258 may be associated with multiple assets, and each asset 259 may be associated with multiple executions (trades) 260. One skilled in the art will readily understand the structure and function of the database schema illustrated in Figure 3.

Figure 4 illustrates a graphic interface permitting a user to *balance* the contents of an Individual Fund™. By using a pointing device such as a "mouse" 280 to move a visual pointer 282 to a bar 284 representing a particular fungible good (e.g. a stock) in the basket on the display 286, and by "dragging" the bar 284 out or in, the remaining bars 288, 289, 290, each representing the other constituents of the particular basket, automatically move proportionally in response to
30 keep the fund total value 292 close to the "target value" 294. For example, if the user wished to

increase the holdings of MSFT 298 and so indicated this by dragging the bar 288 to the right, each of the remaining holdings in the basket would be reduced an amount to offset the increased expense of purchasing shares of MSFT 298. Since the cost of units of each constituent (e.g. shares of stock) is different for each holding, the changes in the number of shares for each of the changed constituents is different for each one. In this example, an increase of MSFT 298 from 20% to 40% would require reducing the number of shares of DELL, INTC and CPQ enough to maintain approximate equal dollar values of each but reducing the total value of those holdings to 60% of the basket (IF™). Weightings can be dollar-weighted, cap-weighted or user defined.

Figure 5 represents an example of an e-mail IF order confirmation. The sample confirmation contains the following information:

- Name and address of executing broker 300;
- customer name 302;
- account number 304;
- IF Name, provided by the user (in this case "Randy's Internet Basket") 306;
- unique IF identification number 308;
- transaction type (i.e. buy or sell) 310;
- total shares of all stocks in basket 312;
- cost of the basket 314;
- date and time order was submitted 316;
- Net Asset Value 318;
- summary of all orders 320;
- detail of each trade 322;

It will be noted that a particular order may be filled with multiple trades. For example, the order for RNWK 324 for 4,800 shares was filled in two trades, one for 1,500 shares 324, and one for 3,300 shares 326. A variety of other and additional information and analysis may be included in this type of report. Note that a "buy" transaction for a basket may include a mix of buying long and selling short. Similarly, a "sell" transaction for a basket may include a mix of selling long and selling short. Therefore, the terms "buy" and "sell" may be considered arbitrary in relation to an IF order, but are used for simplicity of communicating with the user.

In its preferred embodiment, the present invention has been designed and modeled using the Unified Modeling Language ("UML") a standard developed by Rational Software Corporation (<<http://www.Rational.com>>). In addition, an Interface Definition Language Compiler is used to compile the system interfaces. (Any of a multitude of alternative software products or implementations may be freely substituted by the practitioner to achieve the same or equivalent results). Specifically the developer builds an IDL source file which is compiled to

generate a proxy/stub DLL. The proxy acts like the server object in the client process and implements the servers interface. When the client calls the interface of the proxy the proxy packages the call parameters and transmits them across the network to the stub in the server process. The stub unpacks the data and calls the server as if the client were local. This provides location independence. The client may be local or remote (using a proxy stub).

Internet Implementation

One of the most useful implementations of the present invention is providing individual investors the opportunity to create their own personalized baskets (IFs™) by using their own judgment of one of any number of filters or decision-making tools at the front-end. In the preferred embodiment of the present invention a user may utilize a program such as a specialized stock tracking and analysis program (e.g. Trade Station®), a spreadsheet (e.g. Excel®) or other specialized software, or information obtained from stock price analysis services to create *ad hoc* “baskets” of stocks and other securities. For example, a user may wish to purchase only the top twenty best performing S&P 500 stocks. Additionally, the user may wish to weight those stocks paying above a threshold dividend on a certain date fifteen percent higher than the other stocks meeting that threshold. By entering these criteria, the user may then execute a purchase of a “basket” of individual stocks totaling a specified order amount. If, for example, the customer wishes to invest ten thousand dollars, the system would purchase stocks from one or more of the various sources in quantities calculated to total close to ten thousand dollars, but weighted according to whatever weighting criteria the customer may specify.

EXAMPLES

To understand the function and advantage of the present invention, several examples are presented.

Example 1: Manual Basket Trading Without Use of the Invention, Unweighted.

Assuming an investor wished to trade “baskets” of securities without the use of the present invention, he would have to go through the following steps.

- (1) choose a number of individual stocks to purchase;

- (2) decide total amount to invest;
- (3) look up quotes for each stock;
- (4) calculate the number of shares of each stock required to total proposed investment while equally distributing amount invested in each stock;
- 5 (5) log onto Internet stock trading service;
- (6) enter symbol and number of shares of each stock individually;
- (7) enter "buy," "sell," "sell short";
- (8) execute;
- (9) repeat steps 6-8 for each individual asset;
- 10 (10) receive individual confirmations for each asset;
- (11) compare confirmed prices with original quoted prices to determine variation;
- (12) enter all values in spreadsheet program and calculate Net Asset Value (NAV);
- (13) track group of stocks purchased by periodically looking up each stock and enter in spreadsheet;
- 15 (14) Repeat steps 5-10 to close position at a later date.

It may be readily seen, that the above manual method makes it impossible to trade groups of stocks (or other assets) quickly enough to make it practical to manage individual "baskets" of assets. Prices constantly fluctuate so that by the time a stock (or other security) is sold or purchased, its price has changed from the time a quote was obtained. It is further impossible under such circumstances to continually track the NAV of each "basket." Prior-existing tools do

20 not promote thinking at a high level of abstraction. These tools focus on individual trades.

Adding various weightings for each stock creates even higher levels of problems in managing such an *ad hoc* fund. The present invention solves all of these problems.

Example 2: Automated "Basket" Trading Using Present Invention, Unweighted.

- 25 Carrying out the same purchases as in Example 1 using the present invention illustrates the significant contribution made by the claimed invention.
- (1) investor picks a number of individual stocks to purchase;
 - (2) user enters each stock symbol into front-end form (may be carried out as single operation such as using "cut/paste");
 - 30 (3) user enters total amount to invest

- (4) user clicks "preview";
- (5) system looks up the instantaneous market prices of each asset
- (6) system computes allocation of shares to yield equal investment in each stock;
- (7) user clicks "execute";
- 5 (8) confirmation sent to user organized in same way as order, showing NAV for "basket";
- (9) user clicks "close" to view closeout preview;
- (10) user clicks "execute" to close IF.

The execution and confirmation are essentially instantaneous. If the investor wishes to

10 sell the "basket" she created in Example 2, it is simply a matter of choosing the "basket" from the user's on-screen menu and clicking "close." The terminology used here is Open and Close. Opening a position means to buy or sell. Closing a position is to execute the contra trade thus liquidating. Again, the transaction and confirmation is essentially instantaneous. It may be seen that a "basket" of stocks may be created and sold without first owning the stocks making up the

15 "basket." This is known in the art as "selling short." As long as the investor has a margin account, this is permitted. However, the investor may create a "basket" containing stocks in companies A, B and C while she owns another "basket" containing shares in companies C, D and E. If she sells the first "basket" short, the system may automatically remove sufficient shares of company C from the second basket, unless the investor specifically declines to do so (i.e., selling

20 "short against the box"). In this way, interest charges may be eliminated since the short sale of C offset by adequate shares in another "basket" owned by the user merely implements a simple bookkeeping entry by the system.

Example 3: Automated Basket Trading of Individual Stocks, Weighted.

Commonly, an investor may not want to invest equal amounts in several companies. The

25 present invention allows, therefore, for "weighting" assets. See discussion re: "Basket Balancer" above.

- (1) User picks multiple individual stocks to purchase;
 - (2) User enters weighted percentage for each stock. This may in the form of adding weight to individual shares (e.g., MSFT + 15%, MOB + 0, DIS + 25%, INTEL + 10%) or by
- 30 building a fund by percent. (e.g., MSFT 20%, MOB 15%, DIS 50%, INTEL 15%);

- (3) Enter total investment;
- (4) user clicks "Preview"
- (5) System looks up market prices and computes share allocation according to the weight assigned to each share
- 5 (6) user clicks "Execute;.
- (7) System sends confirmation to user organized in same way as order, computing NAV.

It can be seen that the present invention provides total flexibility for investors. The system also includes a "rebalancing" feature in which the weighting may be freely updated at any
10 time. If the "basket" is used to track a specific index such as the S&P 500 or the Dow Jones Industrials, the "basket" may be rebalanced periodically by buying and selling shares to re-weight the IF, tracking changes in the particular index being followed. This allows the IF to continually track the ever-changing composition of the index while maintaining the IF's current market value.

15 **Example 4: Short Sale.**

The present invention may be used to transact short sales – that is, selling shares not yet owned.

- (1) User creates "basket" of stocks;
- (2) User sells "basket" using invention;
- 20 (3) credits user's account with sale proceeds and reserves 150% of current market value ("CMV") for future buy-back.

Example 5: Sell Short "Against The Box."

As touched upon above, there are circumstances where an investor may not wish to sell shares held in one "basket" even though she wishes to sell the same stock short in another
25 "basket." While executing the short sale from the owned asset "basket" may save interest charges, such cross-filling may not be desirable in such circumstances as when the shareholder wishes to retain certain voting rights. Therefore, she may wish to "sell short against the box." In such a circumstance, the seller merely needs to indicate in a short sale that a particular stock or group of stocks should be sold against the box. Such a choice may be implemented merely by

clicking on a box or entering a code designation associated with each stock to be sold against the box. Once entered as such a sale, the system will not cross-fill the order between "baskets."

Example 6: Track Known Index.

Index trading has become popular in recent years. Certain funds have been constructed to
5 "track" indexes such as S&P 500®, Dow Jones® Industrials, NASDAQ® 100 and the like. The present invention permits the investor to track whatever index or indices she wishes and to freely adjust such index-tracking "basket."

In addition, many quantifiable and systematic trading strategies have recently been popularized, such as "DOGS OF DOW", CANSLIM™, VALUELINE®, etc. Any of these
10 "strategies" may be easily implemented using the present invention. The user can react quickly and fluidly to constant changes in market conditions and modify his or her strategy at will.

Since many such indices are widely published, the present invention may incorporate automatic rebalancing for "baskets" tracking particular indices. For example, assume a trader wishes to track the 20 best performing stocks in the Dow Jones Industrials:

- 15 (1) User chooses "track index" option;
- (2) User chooses "Dow Jones Industrials" from index list;
- (3) User picks "Best 20" from menu;
- (4) User enters total amount to invest;
- (5) User chooses "Auto Balance Weekly";
- 20 (6) User presses "execute";
- (7) System looks up Dow Jones Industrials index and determines which stocks are the 20 best performing stocks;
- (8) System allocates share purchases to each stock to invest equal amounts in each stock;
- 25 (9) System executes purchases;
- (10) System sends confirmation in form of "basket" or "Individual Fund™";
- (11) System trades shares weekly automatically to maintain close track with index.

Note that the investor could choose more or less frequent auto-balancing or may only balance when he or she specifically chooses.

All of the above scenarios using the present invention may include automatic stop loss settings and automatic "basket" reporting via e-mail, tele-facsimile, telephone interactive voice response, Internet web pages and the like.

Index-based mutual funds regularly fail to trade indices accurately, commonly returning
5 several percent less than the raw index, and further impact the investor with loads and fees. The present invention permits a trader to track any index he or she wishes with no management fee, paying only a commission for stock trades.

User Interface and Architectural Considerations

The user interface may take many forms and use many different paradigms including a
10 spreadsheet form or graphical interface. Institutional traders may interface the system via the Internet (World Wide Web interface) or through custom interfaces using the Internet or proprietary links such as dedicated lines, satellite links, radio transmission systems, fiber optic links, and the like. Front end application preferences may be saved in the context of the client application, in a spreadsheet for Excel® or a "cookie" for HTML.

15 Importantly, the system in its best mode of implementation is highly *stateless* in that it does not store any data it does not have to, and no transient data is maintained by core components. All durable data are written to a database but individual transaction context and connection data are not stored on the system. This allows the system to remain connectionless and stateless at the core and scale to support many thousands of users. In this way, execution of
20 the distributed architecture is greatly enhanced. The NAV for each "basket" is computed at the time of purchase such that it equals the amount the user intends to spend, at the time of sale of any or all of the assets in the "basket" or upon inquiry by an authorized "contact." The entire architecture underlying the system is based on the "basket" paradigm. Each "basket" is a group of orders. Each order may be broken down. The system actively goes out and seeks orders to fill
25 "baskets."

The best mode of the system uses a load balancing scheme in which each "basket" order is distributed over a load-balancing cluster. That is, multiple devices (computers) may be operating simultaneously on individual purchases to fill a single "basket" order.

It should be apparent to the reader that many variations of the present invention are
30 possible without departure from the scope of the present invention. For instance, and without

limitation, the present invention may be used to trade individual funds containing stocks, bonds, commodities futures, stock options, mutual funds, precious stones such as diamonds, precious metals, entertainment tickets, travel vouchers, airline seats, hotel rooms, hardware inventory, computer memory chips, interval ownership of real estate ("time share"), shares of real estate
5 holdings, or even specific items of real estate, or combinations thereof. The invention may be implemented for professional as well as non-professional traders alike. The invention may be implemented on world-wide networks such as the Internet or on privately-owned networks. Additionally, as discussed above, a wide variety of computer hardware and software variations may be used to implement the invention.

10 The specific implementations disclosed above are by way of example and for the purposes of enabling persons skilled in the art to implement the invention only. Thus, the scope of this invention should be determined by the appended claims and their legal equivalents, rather than by the examples given.

Claims

What is claimed is:

1. A system for trading fungible goods comprising:

- electronic computer means containing a data storage means, a computing means, a data
5 input means, a data output means; and a data communications means;
means for authenticating user;
means for acquiring real-time price quotations for fungible goods from at least one
relevant market such as stock exchanges, electronic trading networks, and futures exchanges;
input means for user to enter total desired target investment dollar amount;
10 input means for user to enter two or more fungible goods species identifiers such as stock
symbols to indicate choices for purchasing said fungible goods species;
said computing means for comparing said real-time trading prices with said user indicated
choices of fungible goods species and calculating the number of each said chosen fungible goods
species based on predefined rules to comprise a basket of fungible goods close to said desired
15 target investment amount;
means for submitting transaction orders for the calculated number of units of each user-
entered fungible goods species to the relevant market;
means for executing said submitted transaction orders;
means for confirming said transaction of each said chosen fungible goods species;
20 means for recording said fungible goods purchases in a database on said data storage
means;
means for debiting from said brokerage account the required dollar amount to fund the
transaction;
means for generating a report of contents of basket of fungible goods created by said
25 execution of said submitted transactions;
means for providing report of history of user's account.

2. The system of claim 1 further comprising:

- means for the user to vary the contents of said basket by entering additional said fungible
30 goods species identifiers after said submittal of said transaction.

3. The system recited in claim 1 further comprising:

means for permitting the adjustment of the ratios of funds invested in individual fungible goods species by inputting different weights for said individual fungible goods specified.

4. The system recited in claim 3 further including:

5 means for automatically rebalancing said basket by inputting criteria for adjusting weights applied to each species of fungible goods in said basket,
specifying criteria to trigger said rebalancing,
executing one or more transactions to adjust the contents of said basket on conformance with adjusted weights upon the occurrence of said criteria.

10 5. The system recited in claim 4 wherein:

said criteria to trigger rebalancing includes a pre-defined interval of time.

6. The system recited in claim 1 wherein:

said predefined rules are created by the user.

7. The system recited in claim 6 wherein:

15 said predefined rules includes applying specified weights to each species of fungible goods.

8. The system recited in claim 1 wherein:

said predefined rules are chosen from a menu by the user.

9. The system recited in claim 1 wherein:

20 said predefined rules are automatically calculated according to predetermined criteria such as relative historical performance of each said selected species of fungible goods.

10. The system recited in claim 1 wherein:

said data communication means includes a wide area network such as the Internet.

11. The system recited in claim 1 wherein:

25 said fungible goods include at least one of the following: equity instruments, debt instruments, futures, mutual funds, commodities, real property ownership shares, or personal property ownership shares.

12. The system recited in claim 1 wherein:

said fungible goods species identifiers include one or more of the following:

stock symbols, company names, debt instrument symbols, debt instrument names, fund symbols, fund names, commodities symbols, commodities names, real estate investment trust symbols, real estate investment trust names, symbols representing shares in real property, descriptions identifying shares in real property, symbols representing certain personal property, or descriptions identifying certain personal property.

13. In a computer network, a method for trading fungible goods such as securities including the steps of

providing a computer network means having a data storage means and a computational means;

10 providing means for linking said computer network means to trading markets for said fungible goods such as national stock markets;

providing an input means for a user to enter data into said computer network means;

the user inputting into said input means a total target dollar amount to be invested in a basket;

15 the user selecting two or more species of fungible goods such as stocks, options, futures or debt instruments to comprise a basket;

determining the real-time price quotation for each unit of said selected species of fungible goods;

providing means to predefine rules for calculating the contents of said basket;

20 entering predefined rules for calculating the contents of said basket;

determining the number of units of each said selected species of fungible goods to purchase close to said target dollar amount according to said predefined rules;

substantially simultaneously submitting orders for all said determined number of units of each selected species of fungible goods in said basket to the relevant trading markets;

25 receiving confirmations of the transaction of each said selected species of fungible goods;

communicating said confirmations of each said transaction to the user;

recording said transactions in a database;

communicating to the user the results of said transactions resulting from a single said submittal.

14. The method recited in claim 13 further including the step of identifying each basket of said fungible goods selected by the user with a unique identifier.

15. The method recited in claim 13 wherein said fungible goods are shares of securities and further including the step of calculating the net asset value of the basket of fungible goods

5 according to the following formula:

$$\text{NAV} = \frac{T_v}{N}$$

where T_v is the total value of all fungible goods in the basket; and

10 N is the number of all said shares in the basket.

16. The method recited in claim 13 wherein said fungible goods are shares of securities and further including the step of calculating the net asset value of the basket of fungible goods according to the following formula:

15
$$\text{NAV} = \frac{T_v - t_x}{N}$$

where T_v is the total value of all fungible goods in the basket;

t_x is transactional costs or load; and

N is the number of all said shares in the basket.

20 17. The method recited in claim 13 further including the steps of:

providing a means for modifying the weight given to each species of said fungible goods;

applying said modified weight to the said determination of the target number of units of each said commodity to trade.

18. The method recited in claim 13 further including the steps of:

25 providing a means for modifying the weight given to one or more species of fungible goods after said one or more species of fungible goods are purchased;

determining the real-time market trading price of each said unit of said species of commodity;

re-determining the target number of said species of fungible goods in each said basket;

30 calculating changes in the number of units of each species of fungible goods in said basket required to optimize the total value of said basket based upon said modified weights assigned to each species of commodity;

substantially simultaneously submitting to the relevant trading markets all trades required to effectuate said calculated changes;

receiving confirmations of the sale or purchase of each said trade;
communicating said confirmation of each said trade to said user;
recording said completed trades in a database;
reporting the modified contents of said basket to the user.

- 5 19. The method recited in claim 17 wherein the weights assigned are modified by the user.
20. The method recited in claim 17 wherein the weights assigned are modified automatically in accordance with predetermined parameters.
21. The method recited in claim 19 wherein said predetermined parameters are predetermined by the user.
- 10 22. The method recited in claim 20 wherein the user chooses said predetermined parameters from a menu.
23. The method recited in claim 13 further including the steps of:
- the user inputting a command to liquidate said basket of fungible goods;
submitting a closing order for each species of fungible goods in said basket to the
- 15 relevant trading market;
receiving confirmations of the transaction for each species of commodity;
recording said confirmations in a database;
communicating said confirmations to said user;
adjusting an account with the proceeds or losses from each sale;
- 20 providing reports of the status of said account.
24. The method recited on claim 13 wherein said transaction is recorded on said computer data storage means.
25. The method recited on claim 13 wherein said transaction is recorded on said data storage means located remotely from the user's said computational means.
- 25 26. The method recited on claim 13 wherein all said communicated results of said transaction are reported in a single report for each basket.

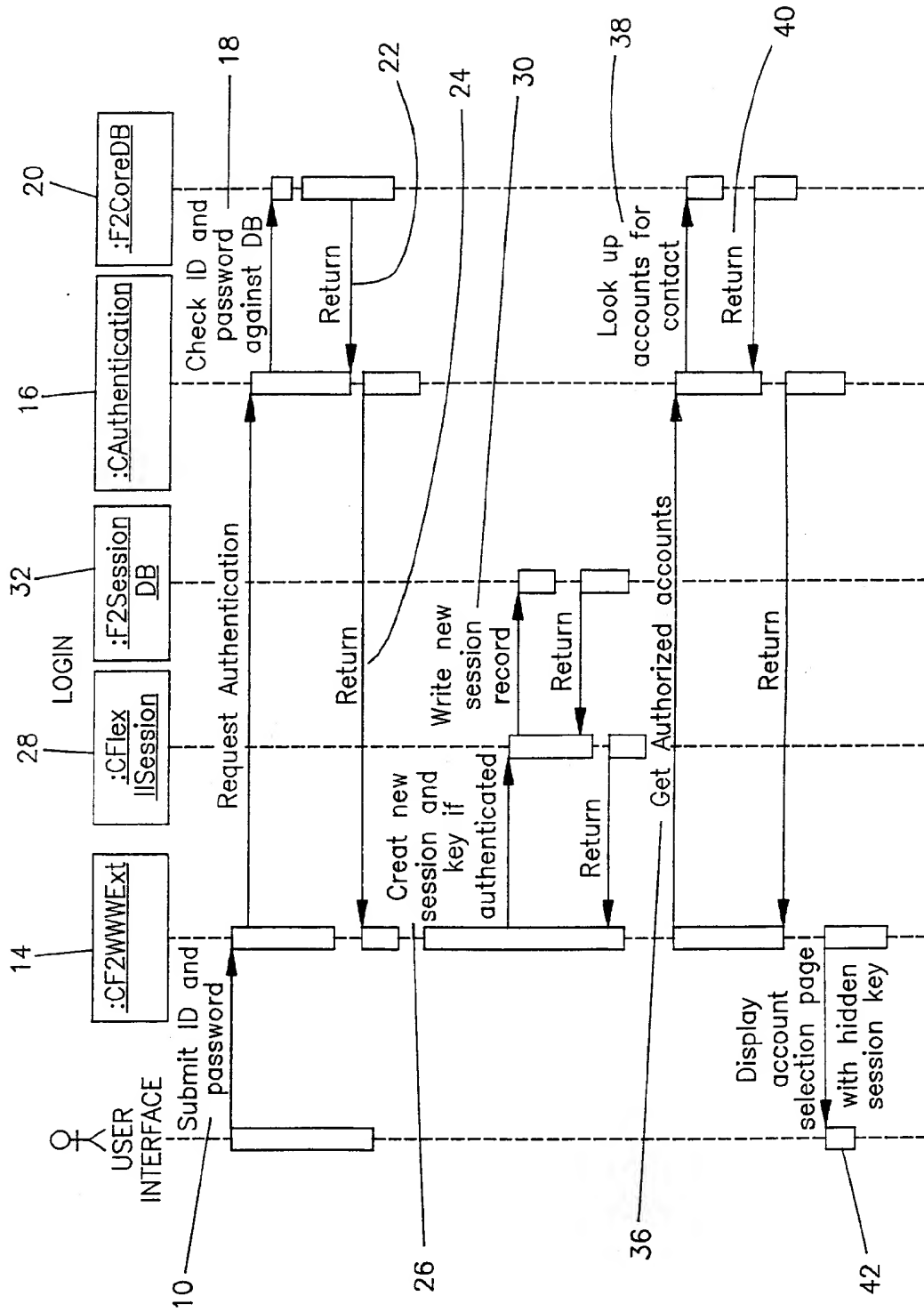


FIG-1A

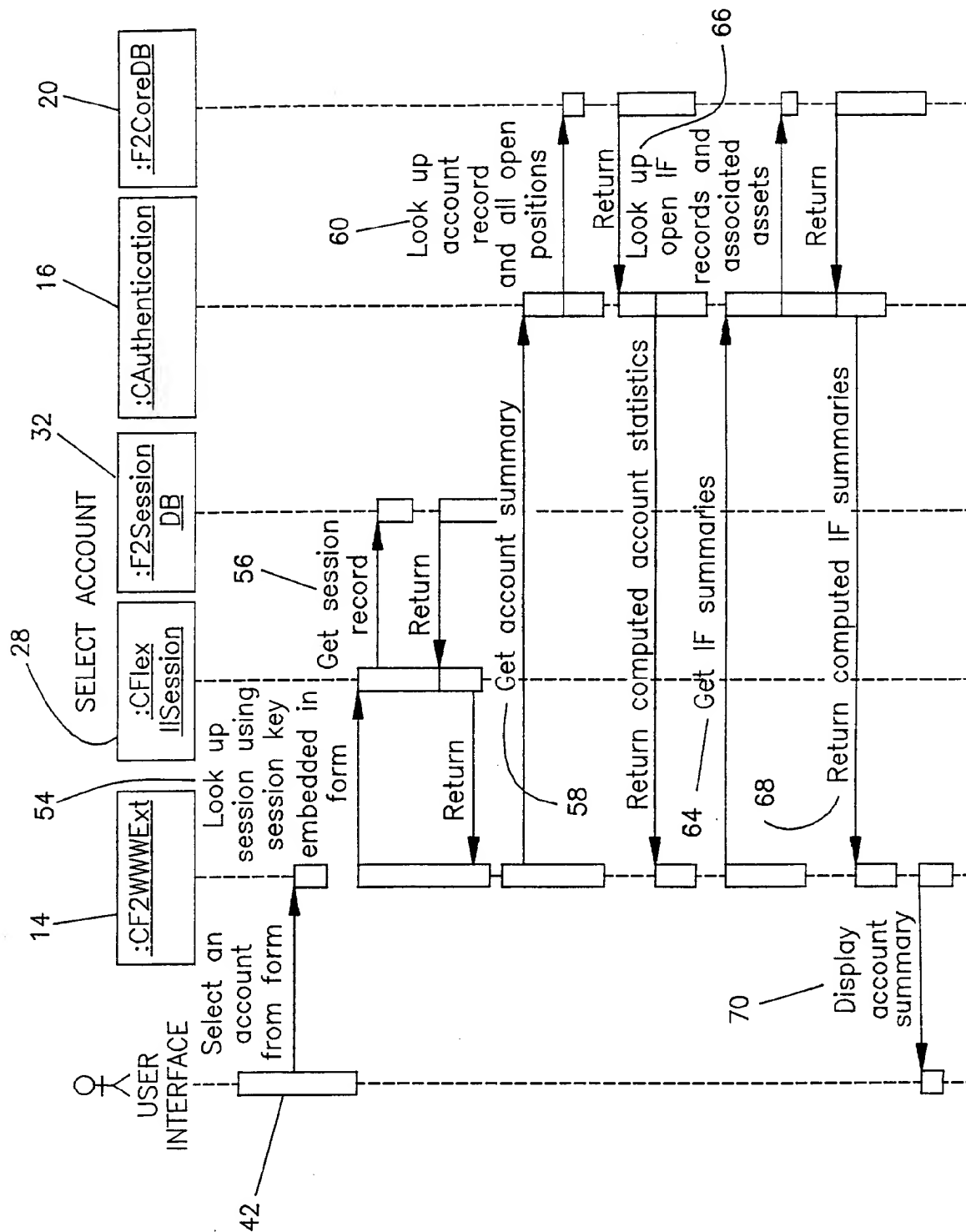


FIG-1B

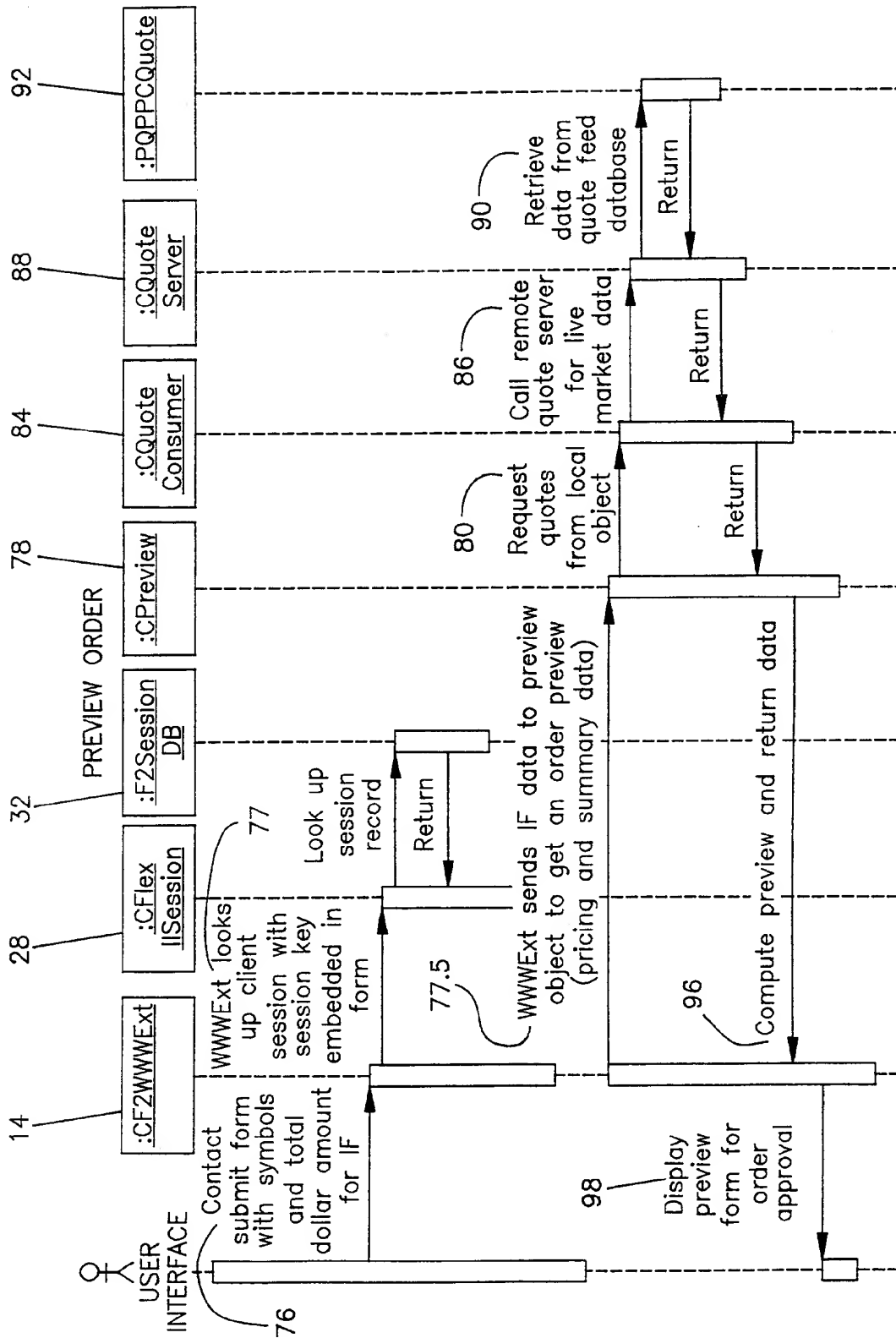


FIG-1C

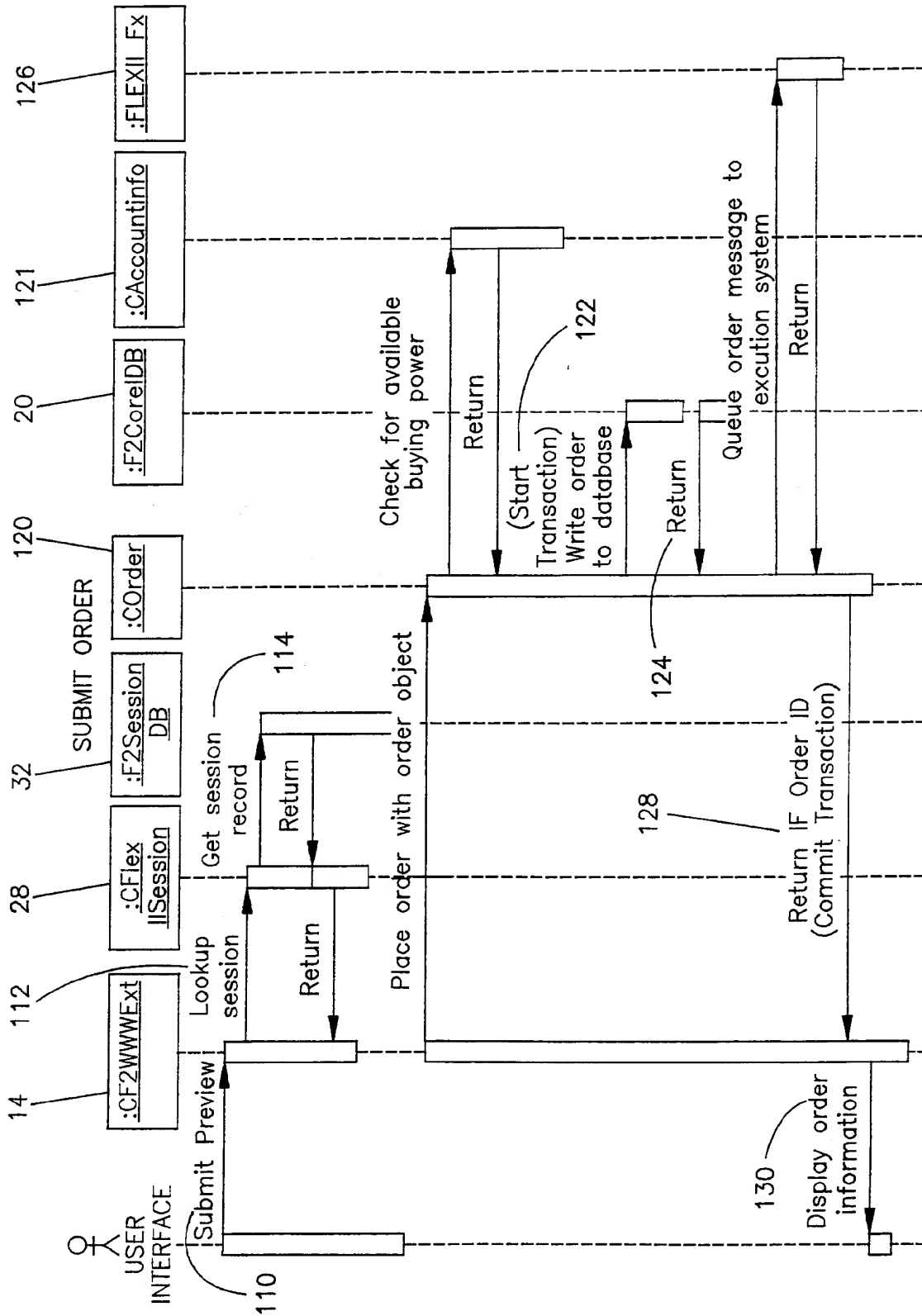


FIG-1D

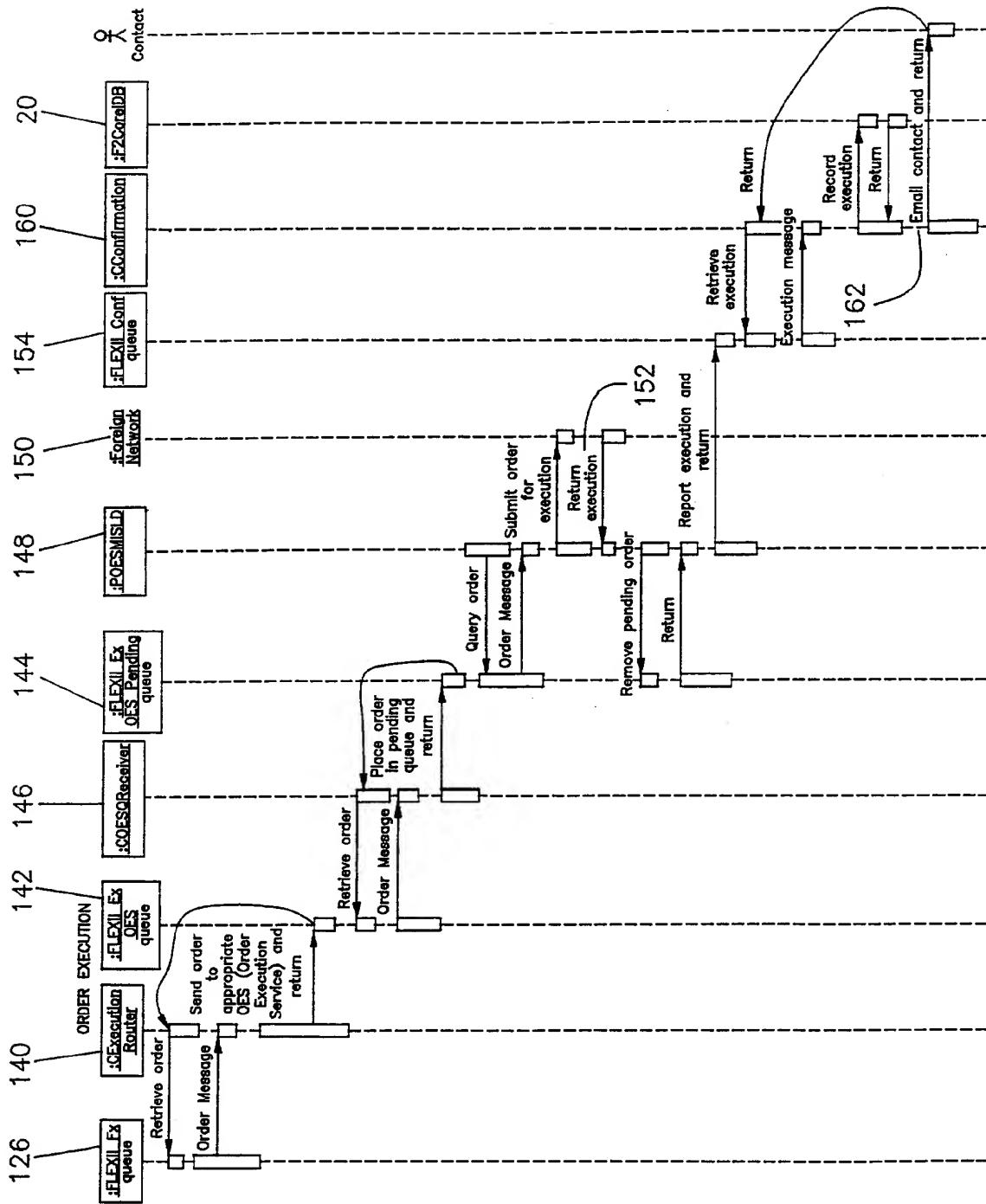


FIG-1E

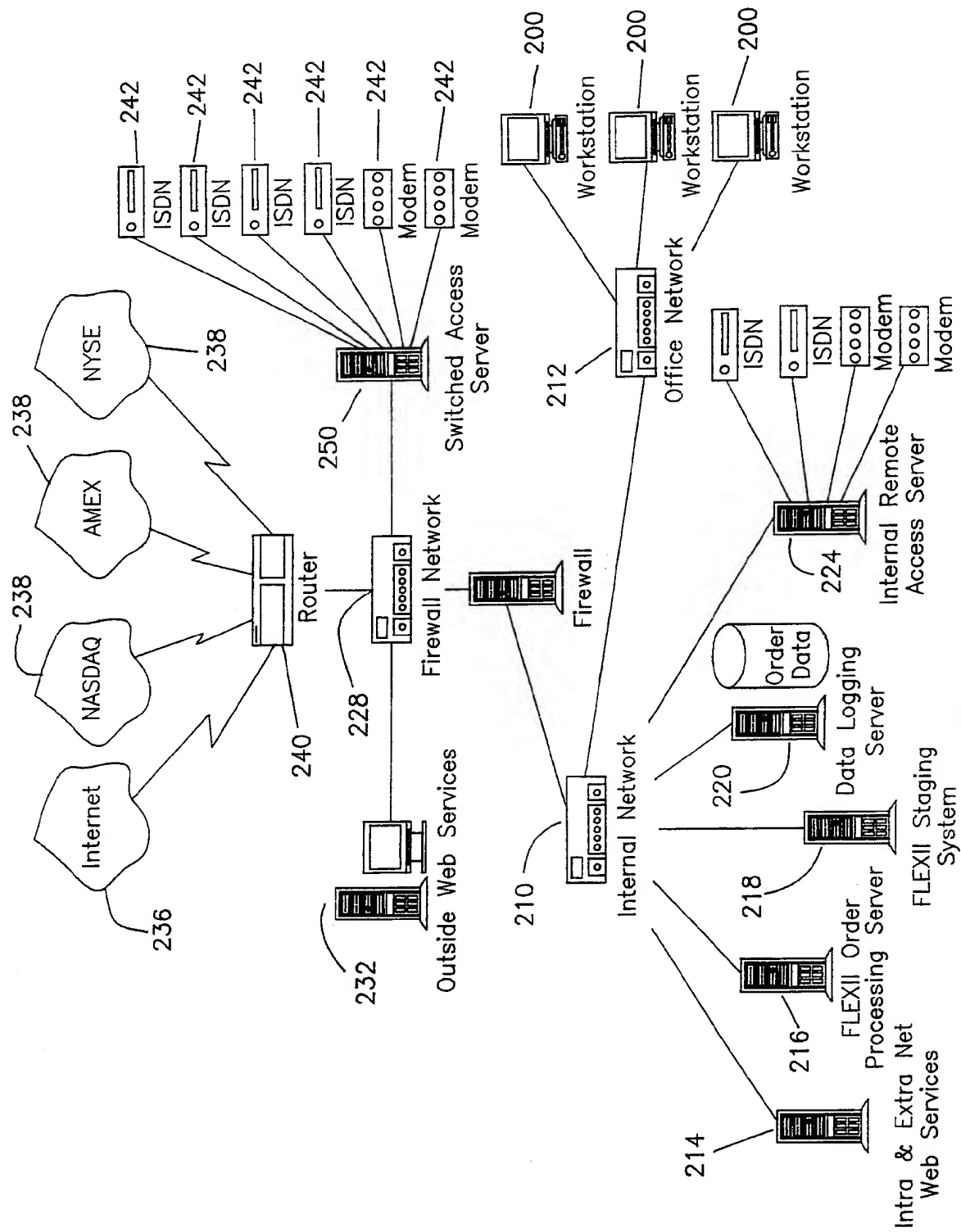


FIG-2

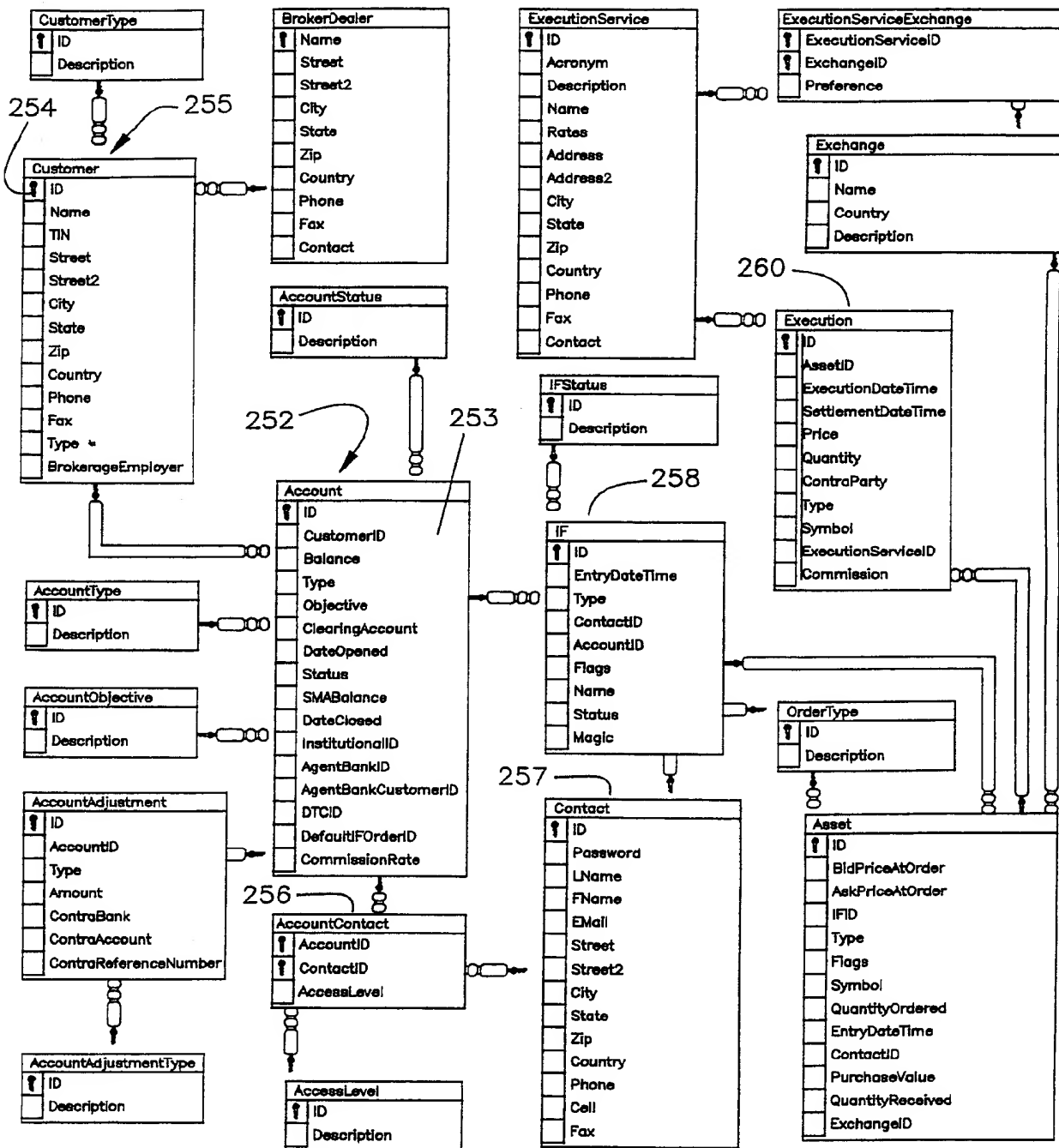
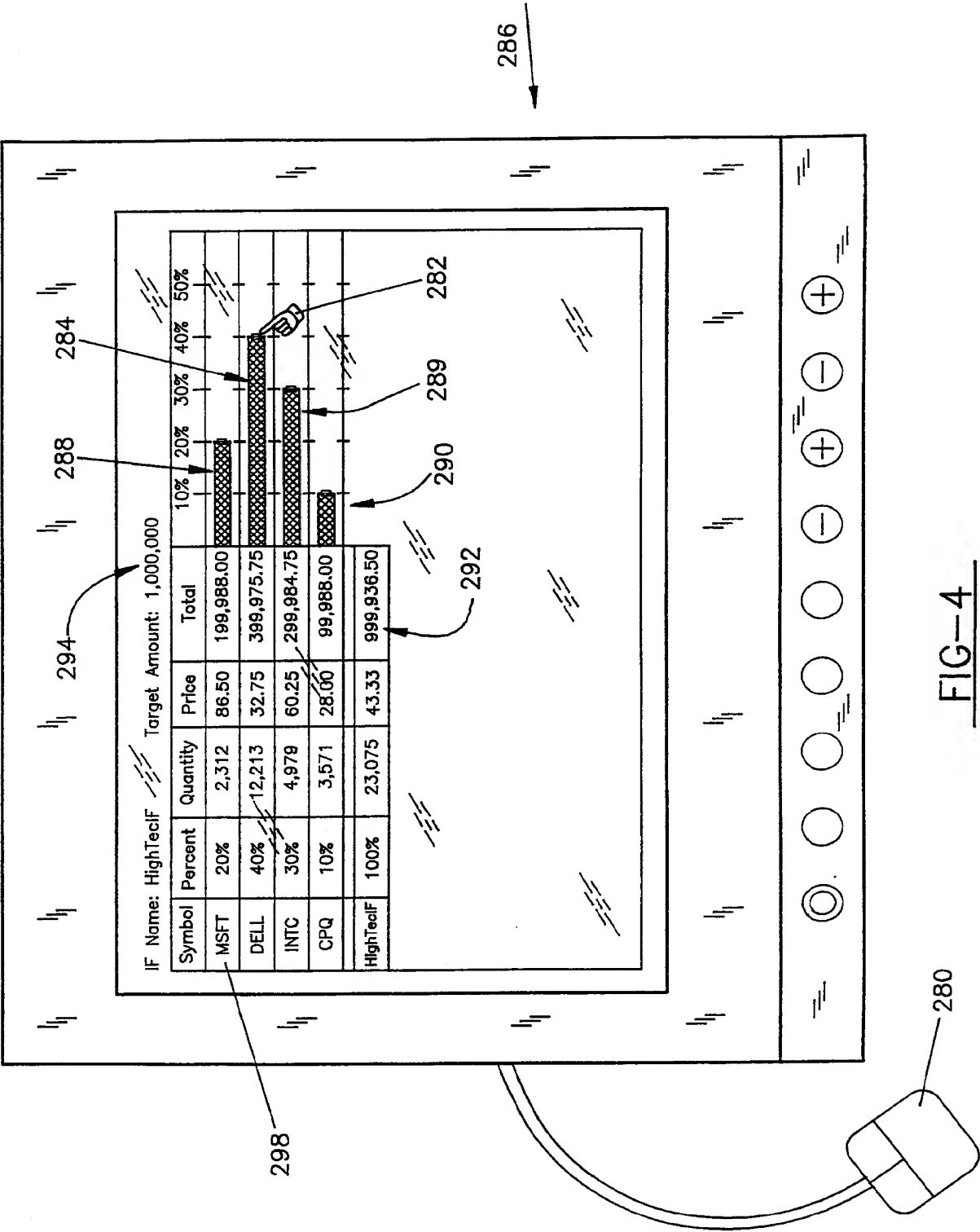


FIG-3



9/9

Universal Network Exchange — 300
 Broker Dealer
 Las Vegas, NV, USA TRADE SUMMARY
 Customer: Abernethy, Randy — 302
 Account: 310 650261 — 304
 IF Name: Randy's Internet Basket — 306
 IF ID: 38456 — 308
 Buy/Sell: Buy — 312
 IF Total Units: 49,700 — 314
 IF Grand Total: 1,351,733.75 — 316
 IF Ordered On: 09/18/98-12:31:12 PDT
 Net Asset Value: 27.1979 — 318
 Orders

Symbol	Name	Units	Avg Price	Commission	Total
AMZN	Amazon.com	1,100	80.8750	80.00	89,042.50
AOL	America Online	1,200	99.2500	80.00	119,180.00
ATHM	At Home	3,000	38.7625	80.00	116,367.50
CKFR	Checkfree Holdings	8,200	10.5549	80.00	86,630.00
ELNK	Earthlink Network	4,000	37.4263	80.00	149,705.00
XCIT	Excite	3,600	26.8750	80.00	96,830.00
SEEK	Infoseek	5,400	20.5810	80.00	111,217.50
LCOS	Lycos	3,600	24.9288	80.00	89,823.75
MSPG	Mindspring Ent.	3,000	34.5000	80.00	103,580.00
NSCP	Netscape Com.	4,400	22.4943	80.00	99,055.00
PSIX	PSINet	7,200	13.7378	80.00	98,992.50
RNWK	Real Networks	4,800	13.7378	80.00	98,992.50
YHOO	Yahoo	1,200	91.6250	80.00	110,030.00

Trades

ID	Symbol	Qty	Price	Total	Trade Date
1	AMZN	1,100	80.8750	88,962.50	9/18/98-12:31:49 PDT
2	AOL	1,200	99.2500	119,180.00	9/18/98-12:31:39 PDT
3	ATHM	1,200	38.5000	46,200.00	9/18/98-12:31:43 PDT
4	ATHM	1,800	38.9375	70,087.50	9/18/98-12:32:01 PDT
5	CKFR	1,000	10.5000	10,500.00	9/18/98-12:31:31 PDT
6	CKFR	7,200	10.5625	76,050.00	9/18/98-12:32:05 PDT
7	ELNK	1,000	37.5000	37,500.00	9/18/98-12:32:01 PDT
8	ELNK	3,000	37.3750	112,125.00	9/18/98-12:32:09 PDT
9	XCIT	3,600	26.8750	96,750.00	9/18/98-12:32:02 PDT
10	SEEK	1,600	20.6250	33,000.00	9/18/98-12:31:29 PDT
11	SEEK	3,800	20.5625	78,137.50	9/18/98-12:31:63 PDT
12	LCOS	500	24.8750	12,437.50	9/18/98-12:31:40 PDT
13	LCOS	3,100	24.9375	77,306.25	9/18/98-12:31:42 PDT
14	MSPG	3,000	34.5000	103,500.00	9/18/98-12:31:36 PDT
15	NSCP	400	22.4375	8,975.00	9/18/98-12:31:27 PDT
16	NSCP	4,000	22.5000	90,000.00	9/18/98-12:31:51 PDT
17	PSIX	700	13.6250	9,537.50	9/18/98-12:31:19 PDT
18	PSIX	6,500	13.7500	89,375.00	9/18/98-12:31:22 PDT
19	RNWK	1,500	24.8750	37,312.50	9/18/98-12:31:28 PDT
20	RNWK	3,300	24.6250	81,262.50	9/18/98-12:31:49 PDT
21	YHOO	1,200	91.6250	109,950.00	9/18/98-12:31:17 PDT

FIG-5